Department of Defense

High Level Architecture

OMT Extensions

Version 1.0

20 August 1996

Table of Contents

FOREWORD ...... **iii**

1.0 PURPOSE **4**

2.0 BACKGROUND **7**

2.1 OMT EXTENSIONS RATIONALE 7
2.2 FEDERATION OBJECT MODELS 7
2.3 SIMULATION OBJECT MODELS 8

3.0 HLA OMT EXTENSIONS COMPONENTS **10**

3.1 COMPONENT STRUCTURE TABLE 11
3.1.1 Purpose/Rationale 11
3.1.2 Table Format 12
3.1.3 Inclusion Criteria 12
3.1.4 Example 14
3.2 ASSOCIATIONS TABLE 14
3.2.1 Purpose/Rationale 14
3.2.2 Table Format 15
3.2.3 Inclusion Criteria 16
3.2.4 Example 16
3.3 OBJECT MODEL METADATA 17
3.3.1 Purpose/Rationale 17
3.3.2 Metadata Format 18

APPENDIX A: TABLE ENTRY NOTATION **21**

ACRONYMS **23**

REFERENCES **25**

List of Tables

TABLE 3-1. COMPONENT STRUCTURE TABLE 13

TABLE 3-2. COMPONENT STRUCTURE TABLE EXAMPLE....................14

TABLE 3-3. ASSOCIATIONS TABLE....................15

TABLE 3-4. ASSOCIATIONS TABLE EXAMPLE....................17

FOREWORD

The formal definition of the Department of Defense High Level Architecture (HLA) comprises three main components: the HLA Rules, the HLA Interface Specification, and the HLA Object Model Template (OMT). These components are documented in the following reports:

- **HLA Rules V1.0**
- **HLA Interface Specification V1.0**
- **HLA Object Model Template V1.0**

This document, the HLA OMT Extensions, is intended to extend the template, described in the HLA Object Model Template V1.0 document, to support additional classes of information relevant to the specification of a particular HLA object model. It is one element in the HLA Technical Library of information sources of general relevance to developing and executing HLA federations. The other elements of the HLA Technical Library that are particularly relevant to HLA object model development include the following:

- **HLA Glossary:** A common set of semantics for terms used in the documents of the HLA formal definition or the HLA Technical Library.

- **HLA Federation Development and Execution Process Model:** A description of the process used to build and execute HLA federations.

- **OMT Use Cases:** A set of case studies describing the process of developing HLA object models in different communities of interest. Each case study is based on the experiences of one of the HLA prototype federations (protofederations).

- **HLA OMT Test Procedures:** A set of procedures for testing compliance of an object model with the HLA OMT.

Other elements of the HLA Technical Library may also have some relevance to HLA object model construction. Users of this document are encouraged to browse the contents of the HLA Technical Library to discover sources of potentially relevant information, and to gain a broader understanding of other general HLA resources.

1. PURPOSE

The Department of Defense (DoD) Modeling and Simulation Master Plan [DOD95] calls for the establishment of a DoD-wide High Level Architecture (HLA) for modeling and simulation (M&S), applicable to a wide range of functional applications. The purpose of this architecture is to facilitate interoperability among simulations and promote the reuse of simulations and their components.

The HLA Object Model Template (OMT) is designed to document HLA relevant information about classes of simulation or federation objects and their attributes and interactions. This document, the HLA OMT Extensions, provides a specification of optional extensions to the HLA OMT. It is intended to define a common template for specifying those classes of information that are not directly supported by the HLA OMT, but that are useful to specify for some M&S applications to facilitate understanding and completeness of the HLA object model. This document is also intended to support the more general goals of the HLA to provide a common framework for development, interoperation, and reuse of simulation systems.

2. BACKGROUND

2.1 OMT Extensions Rationale

A standardized structural framework or template for specifying HLA object models is an essential component of the HLA for the following reasons:

- Provides a commonly understood mechanism for specifying the exchange of public data and general coordination among members of a federation.
- Provides a common, standardized mechanism for describing the capabilities of potential federation members.
- Facilitates the design and application of common tool sets for development of HLA object models.

HLA object models may be used to describe an individual federation member (federate), creating an HLA Simulation Object Model (SOM), or to describe a named set of multiple interacting federates (federation), creating a Federation Object Model (FOM). In either case, the primary objective of the HLA OMT is to facilitate interoperability between simulations and reuse of simulation components. The HLA OMT Extensions support this objective by providing supplementary mechanisms for describing object relationships within an HLA object model, and by providing a means to associate meta-information with federations and/or simulations. All discussion of HLA object models in this document applies to both SOMs and FOMs unless explicitly stated otherwise.

2.2 Federation Object Models

During development of an HLA federation, it is critical that all federation members achieve a common understanding as to the nature or character of all required interactions between participating federates. The primary purpose of an HLA FOM is to provide a specification of the exchange of all public data among federates in a common, standardized format. The content of this public data includes 1) an enumeration of all public object classes, 2) a description of all interaction types and associated parameters, and 3) a specification of the attributes that characterize the public objects. This information is documented using the formats specified in the HLA OMT. In addition, an HLA FOM can include additional information as needed to describe more fully other types of static relationships that may exist between object classes, and to describe key characteristics of the federation as a whole. These optional classes of object model characterizations are described using the formats defined in the HLA OMT Extensions. Taken together, the components of an HLA FOM establish the "information model contract" that is necessary (but not sufficient) to ensure interoperability among the federates.

2.3 Simulation Object Models

A critical step in the formation of a federation is the process of determining the composition of individual simulation systems to best meet the sponsor's overall objectives. An HLA SOM is a specification of the intrinsic capabilities that an individual simulation could offer to potential HLA federations. The standard format in which SOMs are expressed facilitates determination of the suitability of simulation systems to participate in a federation.

The formats described in the HLA OMT and HLA OMT Extensions are generally applicable to either FOMs or SOMs. Thus, SOMs are also characterized in terms of objects and their attributes and relationships. The primary benefit to common utilization of the OMT formats for FOMs and SOMs is that it provides a common frame of reference for describing object models in the HLA community. In some cases, it may even allow SOM components to be integrated as "piece parts" in a FOM, facilitating rapid FOM construction.

3. HLA OMT EXTENSIONS COMPONENTS

HLA object models are composed of a group of interrelated components specifying information about classes of objects, their attributes, and their relationships. While it is possible to represent the information content of these components in many different ways, the HLA requires the documentation of these components in the form of tables. The template for the core of an HLA object model uses a tabular format and consists of the following components:

- **Object Class Structure Table:** To record the subclass-superclass relations between different classes of simulation/federation objects.
- **Object Interaction Table:** To record the types of interactions possible between different classes of objects, their affected attributes and the interaction parameters.
- **Attribute/Parameter Table:** To specify features of the public attributes of objects and the parameters of interactions in a simulation/federation.
- **FOM/SOM Lexicon:** To define all of the terms used in the tables.

Both federations and individual simulations (federates) are required to use all four of the core OMT components when providing an HLA object model, although, in some cases, certain tables may be empty. In some types of HLA applications, it may also be necessary to include supplemental information about certain types of static relationships that can exist between public object classes. This supplemental information is not utilized by the HLA Runtime Infrastructure (RTI) during federation executions, but is useful for some types of applications since it can improve the semantics of the overall object model. In addition, federations and federation members may want to specify certain types of meta-information in their FOMs/SOMs to facilitate reuse in future applications. To support the specification of these types of optional information classes, the HLA OMT Extensions consists of the following components:

- **Component Structure Table:** To record the “part-whole”, or composition relationships between public object classes.
- **Associations Table:** For other associations of interest, apart from component associations.
- **Object Model Metadata:** To specify the characteristics of the federation or simulation.

The basics of these components are presented in the following subsections, along with a brief review of the rationale for including the component in an HLA object model. The template format for each table component is also provided and described. In addition, some criteria are suggested to help guide decisions on when to use these components for a specific HLA object model.

3.1 Component Structure Table

3.1.1 Purpose/Rationale

A component structure for an HLA object model is a structure of part-whole relations between classes of objects from the simulation or federation domain. A part-whole relation between two object classes indicates that objects from one class are parts (or components) of composite objects from another class. Typical part-whole relations include those between the assemblies and subassemblies of machines, as well as inclusion relations between parts of an organization, such as the divisions in a corps or the brigades in a division. An HLA component structure is simply a set of these part-whole (or component) relations along with their cardinalities.

Component structures composed of part-whole relations commonly form trees since individual objects are often conceived as being parts of only a single larger entity. Alternative structures are possible, however, when objects are components in multiple distinct composites.

Part-whole relations have a cardinality or multiplicity, which specifies how many parts of the specified type are included in a composite of the specified type. Since such multiplicity values are ordinarily specified for the classes of such a relation, not the specific instances, they may consist of a set of values to indicate the range over which specific instances of the relation may vary. Although cardinality specifications are optional in the component structure of an HLA object model, it can facilitate comprehension of component structures if they are included. Ordinarily, there are well-known limits on such cardinalities even if there is no precise number for all cases.

The primary purpose of the Component Structure Table is to aid human understanding of object class relationships within a federation or simulation. Component structures can be helpful for communicating a clear understanding of the relationship between composite objects and their components. The components of a composite physical object commonly maintain highly constrained spatial relations with each other, whether in the fixed geometries of many mechanical assemblies or in the more loosely arranged components of an organization. Organizational composites, in particular, may be important for understanding command and control dynamics of simulations that incorporate them.

Component structures may aid in conveying the level of detail of a simulation by describing the depth to which parts are modeled. Information on the depth of component structures can be especially relevant for evaluating the suitability of reuse of a simulation for applications in engineering design and test.

Part-whole relations are just one type of association which are given special attention due to their widespread importance in understanding many systems. This special status is widely recognized in existing object-oriented (OO) software development methodologies where distinctive links are specified for these relations in object model diagrams. Other types of associations may also be included in an HLA object model, as described in Section 3.2, “Associations Table.”

3.1.2 Table Format

The component structure template of Table 3-1 provides a format for representing the component structure, or part-whole hierarchy, of objects involved in a federation or simulation. It begins in the left column with those object classes having the highest level of composition, not being components of any relevant composite entity but being composed of distinguishable components themselves. These entities are then decomposed in the next column to their right by their highest level components. Subsequent columns to the right further decompose the composites to their left down to the lowest level of relevant components. The entries in each column (except for the first) may also include a cardinality to indicate the number of components of that type that are aggregated in the composite object in the column to its left. These numbers may be exact or indicate a range. See Appendix A for a brief description of the notation that is used for specifying entries for this table.

3.1.3 Inclusion Criteria

As with all aspects of an HLA object model, only those components relevant to interoperability and reuse should be included in the component structure. Such relevance is somewhat subjective in that the component structure is not required by the Runtime Infrastructure (RTI) to support object class attribute subscriptions or interactions. Still, guidelines are possible and are described in the following paragraphs.

When the objects of two object classes participate in a part-whole relation and both object classes appear in the class hierarchy for independent reasons, then their part-whole relation should be documented in the HLA component structure. Such documentation can improve comprehension of the roles played by these classes of objects during an execution.

<table>
<tr><th colspan="4">Component Structure Table</th></tr>
<tr><td rowspan="9"><class></td><td rowspan="4"><class> [<number>]</td><td colspan="2">[<class>] [<number>]</td></tr>
<tr><td colspan="2">[<class>] [<number>]</td></tr>
<tr><td colspan="2">...</td></tr>
<tr><td colspan="2">[<class>] [<number>]</td></tr>
<tr><td rowspan="4"><class> [<number>]</td><td rowspan="2">[<class>] [<number>]</td><td>[<class>] [<number>]</td></tr>
<tr><td>...</td></tr>
<tr><td colspan="2">...</td></tr>
<tr><td colspan="2">[<class]> [<number>]</td></tr>
<tr><td>...</td><td colspan="2">...</td></tr>
<tr><td rowspan="3"><class></td><td rowspan="3"><class> [<number>]</td><td colspan="2">[<class>] [<number>]</td></tr>
<tr><td colspan="2">[<class>] [<number>]</td></tr>
<tr><td colspan="2">...</td></tr>
<tr><td>...</td><td>...</td><td colspan="2">...</td></tr>
</table>

Table 3-1. Component Structure Table

The lowest level of individuals who are members of an organization may or may not be included in a component structure. The OO literature, which commonly distinguishes part-whole relations from other types of associations, is divided over whether the relations between individuals and their organization, such as the individual combatants of a platoon, should be treated as part-whole relations. Rumbaugh, for example, argues against treating employees as parts of a company [RUMB91, p. 58], preferring a *Works-for* relation; while Coad and Yourdon treat clerks as parts of organizations [CDYD91]. Such person-to-organization relations may be included in either the Component Structure Table or in the Associations Table of an HLA object model, depending on the needs and preferences of the object model developers.

Derived part-whole relations, which follow by the transitivity of the part-whole relation, should not ordinarily be explicit elements of the component structure. For example, a component structure that includes component relations between *Missile* and *Avionics System*, and between *Avionics System* and *Guidance System* need not include the derived relation between *Missile* and *Guidance System.*

Naturally, classes of objects that are neither components nor composites should not appear in the Component Structure Table, though they may appear in the Object Class Structure Table, discussed in the HLA OMT V1.0 document. Thus, if a simulation or federation does not support any component decomposition for its objects, then there will be no Component Structure Table in its HLA object model.

3.1.4 Example

A simple example of the use of the Component Structure Table is given in Table 3-2. This example provides an extension to the restaurant example described in the HLA OMT V1.0 document. In this case, the object class "Main_Course" is shown as a composite object with two components, that of exactly one "Entree" and at most two "Side_Dish" components. This composite class may provide a convenient grouping for partially characterizing the "Order" class shown in the OMT V1.0 Object Class Structure Table example.

Component Structure Table	
Main_Course	Entree 1
	Side_Dish 0-2

Table 3-2. Component Structure Table Example

3.2 Associations Table

3.2.1 Purpose/Rationale

Part-whole relations are just one type of association between object classes. Other associations include command and control associations between military command units, as well as usage and operations relations between vehicles and stations for refueling, supply, and maintenance. The associations component of an HLA object model is designed to capture all such other associations which are important for assessments of interoperability and reuse. All such associations are indicative of a lasting relationship between objects, as opposed to the momentary relations of interaction events.

Associations may be binary, between pairs of object classes defined in the HLA OMT V1.0 Object Class Structure Table. More generally, the Associations Table allows associations between any number of object classes to be documented (simply by adding extra columns and their entries).

General associations have cardinalities, just as part-whole relations do, only this more general case allows different cardinalities for each object class, while the composite class in a part-whole relation is assumed to be singular. The cardinality of an object class in an association represents the number of objects of that class associated with each element of its associated object class. The specified cardinality may be a range or set of numbers to indicate the variability in multiplicity for instances of the association.

Like part-whole relations, other associations between object classes in a federation are not required by the RTI for establishing interoperability between simulations. Thus, their inclusion in HLA object models is also intended as an optional aid for understanding of object class relationships within a federation or simulation. Associations are recognized for their central role in object models as described in the literature of OO development methodologies, where they specify information access paths between associated objects.

3.2.2 Table Format

The Associations Table template, illustrated in Table 3-3, is used to represent other types of possible associations between objects in a simulation or federation besides component associations. This table template is composed of a minimum of three columns, two for the classes of objects involved in a binary association and one for the association itself. These columns are shown here with the Association column in the middle as an aid to readability. Each row can be read as asserting the existence of one type of association between objects in the first class and objects in the second class. Additional columns may be added on the right side of the table for higher order associations between more than two objects. See Appendix A for a brief description of the notation that is used for specifying entries for this table.

Associations Table		
First Class	**Association**	**Second Class**
<class> [<number>] [(<role>)]	<association>	<class> [<number>] [(<role>)]
<class> [<number>] [(<role>)]	<association>	<class> [<number>] [(<role>)]
. . .	. . .	. . .
. . .	. . .	. . .

Table 3-3. Associations Table

Cardinality numbers may be included when they are considered relevant. The same cardinality numbering scheme is used here as in the Component Structure Table. Now, however, both object classes in any given association may have cardinalities. Every member of the first object class has the named association with the number of members of the second object class appearing in the second class column. Correspondingly, every member of the second object class has the inverse of the named association with the number of members of the first object class included the first class column. These numbers may be specified as zero or more (0+) to indicate an optional association, or as other ranges (e.g., 1-2) to indicate other variabilities.

Object roles are another optional entry for the object classes of an Associations Table. Object roles designate the roles that objects of the designated class will play when they take a particular position in an association. For example, a command relationship between the Commander of a Joint Task Force (CJTF) and his staff might have the role of *commander* assigned to the CJTF and that of *subordinate* assigned to his staff.

3.2.3 Inclusion Criteria

In nearly all conceivable implementations of an HLA object model, there will exist some associative relationships between object classes that could be captured via the Associations Table. The primary criterion for the inclusion of this type of information in an HLA object model description is whether this information needs to be made explicit to achieve given objectives. For instance, in HLA federations, it may be necessary during FOM construction to formally document associations between public object classes to achieve the degree of interoperability required for that application. In individual simulations, it may be necessary to formally document associations between object classes in the SOM to convey how certain functionalities are being modeled. In both cases, the general guidance is to include the Associations Table in the HLA object model if such information may be pertinent to interoperability and/or reuse in the present or future applications, and adds to the clarity and completeness of the object model.

3.2.4 Example

Table 3-4 provides examples of associative relationships which may be documented via the Associations Table. The object classes shown in these examples are taken from the Object Class Structure Table example described in the HLA OMT V1.0 document. In all cases, the optional cardinality field has been provided. Although the semantics of the associations shown in this example are generally clear, the specification of roles are included in the “Customer-Eats-Food” association for illustrative purposes.

Associations Table		
First Class	**Association**	**Second Class**
Greeter 1	Seats	Customer 1+
Waiter 1	Takes	Order 1+
Cook 1+	Prepares	Food 1 +
Waiter 1	Serves	Food 1 +
Customer 1+ (Consumer)	Eats	Food 1+ (Consumable)
Customer 1 +	Pays	Bill 1+

Table 3-4. Associations Table Example

3.3 Object Model Metadata

3.3.1 Purpose/Rationale

Metadata for an HLA object model consists of information about a federation or simulation as a whole, as contrasted with the more specific information about object classes, attributes, and interactions which are recorded in the components of a FOM or SOM. For simulations, metadata includes information about general operating characteristics, including general hardware and software characteristics or constraints. Such information can prove helpful in determining the potential for interoperability with other federates and/or federations. For federations, metadata is useful for federation developers to determine the potential reuse of all or parts of an existing FOM. Some classes of federation metadata are common with that of simulations, but metadata also includes additional information on the composition and configuration of the federation itself. Although federation/simulation metadata is an optional component of an HLA object model, developers are strongly encouraged to include this information for the reasons stated.

In the next subsection, a brief list of metadata classes are identified in order to provide a common core of information categories for specifying the types of information to accompany HLA object models. If FOM/SOM developers believe that additional metadata (beyond the core metadata) may be useful to prospective users, it may be recorded in the "Other Metadata" category. Additional core metadata categories may be added via feedback from the M&S community, as described on the last page of this document.

3.3.2 Metadata Format

This section defines the categories of metadata identified for HLA federations and federation participants. At the end of each metadata category, an indicator is provided to specify whether the metadata category applies to federations only (F), simulations only (S), or both (FS).

Name: A name for the federation or simulation. (FS)

Class: Virtual, Live, Constructive, or Hybrid. (FS)

Domain: The general type of application domain (e.g., Strike, AirDefense). Might be taken from categories of the Conceptual Models of the Mission Space (CMMS) when available. (FS)

Purpose: The specific purpose for which the federation or simulation was created. Problem description should provide some notion of fidelity requirements for the given application. (FS)

Modification Date: The date on which the accompanying FOM or SOM was last updated, to be the creation date if there have been no updates since then. (FS)

Sponsor: The organization which sponsored the development of the simulation or federation. (FS)

Developer: The primary developer organization for the federation or simulation. (FS)

Point of Contact: A person designated as a point of contact by the sponsoring agency who can handle inquiries on operations and distribution. Includes name of person, telephone number, physical mail, and electronic mailing addresses. (FS)

Federation Participants: Besides the primary developer, other organizations that participated in the federation. Includes the names of all simulations that participated in the federation via these organizations. (F)

Security Characteristics: For federations, describes the security requirements for the given application and associated facilities. For simulations, includes specification of any requirements for operating at a specific minimum security level. May include specifications of encryption device implementation, network security implementation, and tempest implementation. (FS)

VV&A History: Verification, validation, and accreditation history. Documents the type of VV&A performed, by whom, and when. (FS)

Tool Use: A short description of the primary tools that were used in the development and specification of the federation or simulation. (FS)

Key Features: A listing of the key features of the simulation considered most relevant to reuse in federations. (S)

Computational Capacity Capability: Specifies the actual computational capabilities of a default platform on which the simulation ordinarily runs (if any). Includes architecture implementation, CPU characteristics (processors, memory, cache memory, communication, performance), and disk storage capacity. (S)

Configuration Requirements: Requirements for hardware and software for any computer that might host the simulation. May include requirements for CPU, operating system, memory, compilers, and/or ancillary software. (S)

Programming Languages: The names of all the programming languages used in implementing any part of the simulation/federate. (S)

Time Management: Time-management characteristics of the simulation/federate which may include minimum/maximum update rates, and capabilities for time-sliced or event-driven simulation. (S)

Documentation References: References to any available analysis, design, and/or implementation documentation for the federation or simulation beyond its object model. (FS)

Other: May include any other general information considered relevant. (FS)

In all cases, developers are free to utilize whatever categories of metadata that are relevant to characterize their federation or simulation. The general format for this metadata is simply a listing of each category name followed by free text containing the information for that category. A two-column table format, as used in the FOM/SOM Lexicon, is also acceptable.

Appendix A: Table Entry Notation

The OMT and OMT Extensions table specifications use a subset of Backus-Naur Form (BNF) [NAUR60] to specify the types of entries that belong in particular table cells. In BNF, the types of terms to be substituted in the table are enclosed in angle brackets (e.g., *<class>*). Optional entries are enclosed in square brackets. Any parentheses are terminal characters which should appear as shown. For example, the basic entry in a cell of the Object Class Structure Table (ref. HLA OMT V1.0 document), designated by *<class> (<ps>)*, indicates a class name followed by a Publishable/Subscribable code in parentheses. An asterisk (*) is used to indicate a repetition of zero or more instances, such as in the last column of the Object Class Structure Table where it indicates a variable number of entries for the most specific types of classes, as follows:

[<class> (<ps>)] [,<class> (<ps>)]* | [<ref>]

A vertical bar (|) is used to indicate alternative possible entries. Thus, the specification for the last column of the Object Class Structure Table indicates optional entries of either a variable length list of classes with Publishable/Subscribable codes or a reference to another table.

Acronyms

BNF	Backus-Naur Form
CJTF	Commander of a Joint Task Force
CMMS	Conceptual Models of the Mission Space
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
FOM	Federation Object Model
HLA	High Level Architecture
M&S	Modeling and Simulation
OMT	Object Model Template
OO	Object-Oriented
RTI	Runtime Infrastructure
SOM	Simulation Object Model

References

[CDYD91] Coad, P. and E. Yourdon, *Object-Oriented Analysis*, 2nd edition, Englewood Cliffs, Prentice Hall, 1991.

[DOD95] Department of Defense, Under Secretary for Defense (Acquisition and Technology), *DoD Modeling and Simulation (M&S) Master Plan*, Washington, DC, Department of Defense, October 1995.

[NAUR60] Naur, P. et al., "Report on the Algorithmic Language ALGOL 60," *Communications of the ACM*, Vol. 6, No. 1, January 1963, pp. 1-17.

[RUMB91] Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*, Englewood Cliffs, Prentice Hall, 1991.

Comments on this document should be sent by electronic mail to the Defense Modeling and Simulation Office (DMSO) Object Model Template Working Group reflector (omtmplt@msis.dmso.mil). The subject line of the message should include the section and line numbers referenced in the comment. The body of each submittal should include (1) the name and electronic mailing address of the person making the comment (separate from the mail header), (2) reference to the portion of this document that the comment addresses (by section number, page, and line number), (3) a one sentence summary of the comment or issue, (4) a brief description of the comment or issue, and (5) any suggested resolution or action to be taken.